Available online at www.sciencedirect.com

ScienceDirect

IFAC Papers Online CONFERENCE PAPER ARCHIVE

IFAC PapersOnLine 51-15 (2018) 1092–1097

Regularized Moving-Horizon PWA Regression for LPV System Identification ⋆

Manas Mejari * **Vihangkumar V. Naik** * **Dario Piga** ** **Alberto Bemporad** *

* *IMT School for Advanced Studies Lucca, 55100 Lucca, Italy (e-mail: {manas.mejari, vihangkumar.naik, alberto.bemporad}@imtlucca.it).*

** *IDSIA Dalle Molle Institute for Artificial Intelligence SUPSI-USI, 6928 Manno, Switzerland. (e-mail: dario.piga@supsi.ch)*

Abstract: This paper addresses the identification of *Linear Parameter-Varying* (LPV) models through regularized moving-horizon *PieceWise Affine* (PWA) regression. Specifically, the scheduling-variable space is partitioned into polyhedral regions, where each region is assigned to a PWA function describing the local affine dependence of the LPV model coefficients on the scheduling variable. The regression approach consists of two stages. In the first stage, the data samples are processed iteratively, and a *Mixed-Integer Quadratic Programming* (MIQP) problem is solved to cluster the scheduling variable observations and simultaneously fit the model parameters to the training data, within a relatively short moving-horizon window of the past. At the second stage, the polyhedral partition of the scheduling-variable space is computed by separating the estimated clusters through linear multi-category discrimination.

Keywords: Linear parameter-varying models, PWA regression, Mixed-integer programming.

1. INTRODUCTION

Linear Parameter Varying (LPV) models represent a natural extension of *Linear Time Invariant (LTI)* models. The property of linearity is preserved in the dynamic relation between input and output signals but this relation can change over time according to a measurable time-varying signal, the so called *scheduling variable* p.

Identification techniques for LPV systems, both for state-space and input-output representations, can be classified into two major approaches - parametric and non-parametric methods. In parametric LPV identification, see (Bamieh and Giarré, 2002; Piga et al., 2015; Mejari et al., 2018), the scheduling dependencies of the underlying p-dependent coefficient functions are specified a priori in terms of the so-called "basis functions". The selection of the basis functions remains an open problem, which is partly overcome by non-parametric methods (Tóth et al., 2011; Hsu et al., 2008; Piga and Tóth, 2013; Mejari et al., 2016), where the underlying scheduling-variable dependencies are captured via "kernel functions". However, the choice of the kernel and tuning of the kernel hyper-parameters is still a critical issue.

Alternatively, in this paper, we formulate the identification of LPV models as a *PieceWise Affine* (PWA) regression problem. PWA models are simple and flexible model structures having universal approximation properties such that any nonlinear function can be modelled with arbitrary accuracy by a PWA map (Breiman, 1993). In this contribution, the scheduling-variable space is partitioned into polyhedral regions, where each region is assigned to a PWA function describing the local affine dependence of the underlying LPV model coefficients on the scheduling variable. Both the local affine functions and the partition of the scheduling variable domain are directly estimated from data, by properly adapting the regularized moving-horizon algorithm for PWA regression recently proposed by authors in (Naik et al., 2017). This algorithm consists of two stages. At the first stage, a *Mixed-Integer Quadratic Programming* (MIQP) problem is formulated to cluster the scheduling variable observations and to estimate the p-depend local affine functions in a recursive way. In order to solve the formulated MIQP problem, we use *Branch and Bound* (B&B) method combined with the accelerated dual gradient projection (GPAD) (Patrinos and Bemporad, 2014) at its core to solve the relaxed QP subproblems. The prime advantage of the GPAD-B&B (Naik and Bemporad, 2017) is that it is a simple library-free solver requiring only basic arithmetic operations having comparable performance with the commercial solvers like GUROBI for small-scale problems. At the second stage, the polyedral partition of the scheduling-variable domain is obtained by separating the clusters (estimated at the first stage) by using the multi-class linear separation methods proposed in (Breschi et al., 2016). The paper is organized as follows. In Section 2, the LPV identification problem is formulated in terms of PWA regression, which is solved through the regularized moving-horizon algorithm discussed in Section 3. Two numerical examples on the identification of SISO and MIMO LPV systems are presented in Section 4.

⋆ This work was partially supported by the H2020-723248 project "DAEDALUS - Distributed control and simulation platform to support an ecosystem of digital automation developers".

10.1016/j.ifacol.2018.09.048

2. PROBLEM SETTING

In this section, we formulate the LPV system identification problem in terms of PWA regression. Let us consider the LPV-ARX model structure

$$y(k) = a_0(p(k)) + \sum_{j=1}^{n_a} a_j(p(k))y(k-j) + \sum_{j=1}^{n_b} a_{j+n_a}(p(k))u(k-j) \quad (1)$$

where $p(k) \in \mathcal{P} \subseteq \mathbb{R}^{n_p}$ is the measurement of the scheduling signal at time k, $u(k) \in \mathbb{R}^{n_u}$ and $y(k) \in \mathbb{R}^{n_y}$ are the model input and output vectors, respectively.

Our goal is to fit PWA p-dependent coefficient functions $a_j(p) : \mathbb{R}^{n_p} \to \mathbb{R}^{n_y}$ (with $j = 0, \dots, n_a + n_b$) to a given training dataset $\mathcal{D}_N = \{u(k), y(k), p(k)\}_{k=1}^N$ of length N. Each coefficient function $a_j(p)$ is parametrized by the PWA map:

$$a_j(p) = \begin{cases} \Theta^0_{1,j} + \sum_{h=1}^{n_p} \Theta^h_{1,j} p_h \text{ if } p \in \mathcal{P}_1, \\ \vdots \\ \Theta^0_{s,j} + \sum_{h=1}^{n_p} \Theta^h_{s,j} p_h \text{ if } p \in \mathcal{P}_s, \end{cases} \quad (2)$$

where p_h denotes the h-th element of vector p, $s \in \mathbb{N}$ is the number of modes, (i.e., the number of affine local functions defining a_j), $\Theta^h_{i,j}$ $(h = 1, \dots n_p)$ are parameter matrices of proper dimension, and $\mathcal{P}_i$, with $i = 1, \dots, s$, are polyhedra that form a complete polyhedral partition [1] of the scheduling-variable domain $\mathcal{P}$. In order to introduce flexibility in the LPV model (1)–(2), the polyhedral partition $\{\mathcal{P}_i\}_{i=1}^s$ is not fixed a priori and will be directly reconstructed from data. This represents one of the main advantages with respect to widely used parametric LPV identification approaches which need parameterization of $a_j(p(k))$ as a linear combination of some known basis functions (e.g. polynomial functions).

Let us now stack the parameter matrices as:

$$\Theta_i = \begin{bmatrix} \Theta^0_{i,0} & \cdots & \Theta^{n_p}_{i,0} & \cdots & \Theta^0_{i,n_a+n_b} & \cdots & \Theta^{n_p}_{i,n_a+n_b} \end{bmatrix}, \quad (3)$$

for all modes $i = 1, \dots, s$, and let us introduce the regressor vector $x(k)$

$$x(k) = \begin{bmatrix} 1 \\ y(k-t) \\ \vdots \\ y(k-n_a) \\ u(k-1) \\ \vdots \\ u(k-n_b) \end{bmatrix} \otimes \begin{bmatrix} 1 \\ p(k) \end{bmatrix}, \quad (4)$$

where $\otimes$ denotes the Kronecker product. Substituting (2) into (1), and based on the above notation, the LPV model (1) can be written in the compact PWA-LPV form

$$y(k) = \begin{cases} \Theta_1 x(k) \text{ if } p(k) \in \mathcal{P}_1, \\ \vdots \\ \Theta_s x(k) \text{ if } p(k) \in \mathcal{P}_s. \end{cases} \quad (5)$$

The estimation of the PWA-LPV model (5) consists of: (i) selection of the number of modes s; (ii) estimation of the model parameter matrices Θ_i; and (iii) computation of the polyhedra $\mathcal{P}_i$ defining the partition of the scheduling variable space $\mathcal{P}$. In the rest of the paper, we assume that the number of local affine models s is fixed by the user, and chosen via cross-validation.

[1] A collection $\{\mathcal{P}_i\}_{i=1}^s$ is a complete partition of the space $\mathcal{P}$ if $\bigcup_{i=1}^s \mathcal{P}_i = \mathcal{P}$ and $\mathring{\mathcal{P}}_i \cap \mathring{\mathcal{P}}_j = \emptyset$, $\forall i \neq j$, with $\mathring{\mathcal{P}}_i$ denoting the interior of $\mathcal{P}_i$.

3. PWA REGRESSION ALGORITHM

The algorithm for PWA regression recently proposed by authors in (Naik et al., 2017) is properly adapted to fit the PWA-LPV model (5) to the training dataset $\mathcal{D}_N$. The two stages of the algorithm are as follows:

S1. Recursive *estimation* of the model parameters Θ_i, $i = 1, \dots, s$, and simultaneous *clustering* of the scheduling variables $\{p(k)\}_{k=1}^N$.

S2. *Computation of a polyhedral partition* of the scheduling variable space $\mathcal{P}$ using multi-category linear separation methods.

3.1 Recursive clustering and parameter estimation

The regularized moving-horizon identification algorithm is implemented in stage **S1**. The training data samples $\{x(k), y(k), p(k)\}$ are processed iteratively. At each time sample k, a moving-horizon window of length T containing training samples $\{x(k), y(k), p(k)\}$ from time $k - T + 1$ to time k is considered. The model parameters Θ_i and the active mode $\sigma(k) \in \{1, \dots, s\}$ at time k are estimated simultaneously by solving the mixed-integer programming problem:

$$\min_{\substack{\{\Theta_i\}_{i=1}^s \\ \{\delta_i(k-t)\}_{i=1,t=0}^{s,T-1}}} \sum_{i=1}^{s} \sum_{t=0}^{T-1} \|(y(k-t) - \Theta_i x(k-t))\, \delta_i(k-t)\|^2 \quad (6a)$$

$$+ \sum_{t=1}^{k-T} \left\| y(t) - \Theta_{\sigma(t)} x(t) \right\|^2 \quad (6b)$$

$$+ \sum_{t=0}^{T-1} \sum_{i=1}^{s} \|(p(k-t) - c_i)\delta_i(k-t)\|^2 \quad (6c)$$

$$\text{s.t. } \delta_i(k-t) \in \{0,1\}, \ \sum_{i=1}^{s} \delta_i(k-t) = 1, \ t = 0, \dots, T-1. \quad (6d)$$

The scheduling vector $p(k)$ is assigned to a cluster $\mathcal{C}_{\sigma(k)}$ [2] (having centroid c_i) based on the active mode $\sigma(k) \in \{1, \dots, s\}$ obtained from the optimizer of problem (6), i.e.,

$$\sigma(k) = i^*, \quad \text{with } \ i^* : \delta_{i^*}(k) = 1; \quad (7)$$

According to a moving-horizon estimation strategy, only the active mode $\sigma(k)$ at time k is kept, and the T-length time window is shifted forward to process the next pair $\{x(k+1), y(k+1), p(k+1)\}$.

Note that, the aim of problem (6) is to obtain the optimal sequence of active modes within the T-length horizon and the model parameters Θ_i which best fits the data measured up to time k. The objective of the term (6a) is to simultaneously find the model parameters and the sequence of active modes $\{\sigma(t)\}_{t=k-T+1}^k$ which best match the observations within the T-step time horizon. The term (6b) takes into account the time history of the observations outside the considered time window. Note

[2] The cluster $\mathcal{C}_i$ corresponds to the polyhedron $\mathcal{P}_i$ for $i = 1, \dots, s$.

that, in (6b), the sequence of active modes is not optimized from time 1 to time $k-T$, but it is fixed to the estimates $\{\sigma(t)\}_{t=1}^{k-T}$ obtained from the previous iterations of the moving-horizon estimation algorithm. The sequence of active modes is optimized only within the considered time horizon in (6a) and in (6c). The term (6c) penalizes the distance of the scheduling vector $p(k)$ from the centroid c_i of the cluster $\mathcal{C}_i$. Overall, the active mode $\sigma(k)$ is selected based on the trade-off between the fitting error term (6a) and the penalty on the current scheduling sample $p(k)$ from the centroids of each clusters in (6c). A trade off between complexity of the optimization problem (6), and accuracy in model parameters Θ_i estimate and cluster assignment of $p(k)$ is achieved by varying the length T of the horizon.

Recursive update of the objective function

We point out that, the regularization cost (6b) can be recursively updated once a new observation is available at time k, without requiring to store the entire time-history of observations outside the T-length window.

Let us rewrite the regularization term (6b) as

$$\sum_{t=1}^{k-T}\mathrm{tr}\Big(\big(y(t)-\Theta_{\sigma(t)}x(t)\big)\big(y(t)-\Theta_{\sigma(t)}x(t)\big)'\Big)= \mathrm{tr}\left(\sum_{t=1}^{k-T}\Theta_{\sigma(t)}x(t)x(t)'\Theta'_{\sigma(t)}\right)- 2\mathrm{tr}\left(\sum_{t=1}^{k-T}\Theta_{\sigma(t)}x(t)y(t)'\right)+\mathrm{tr}\left(\sum_{t=1}^{k-T}y(t)y(t)'\right), \quad (8)$$

with $\mathrm{tr}(\cdot)$ denoting the matrix trace.

Let us now define the matrices

$$H_i(k-T)=\sum_{t=1}^{k-T}x(t)x(t)'h_i(t), \quad (9a)$$

$$F_i(k-T)=\sum_{t=1}^{k-T}x(t)y(t)'h_i(t), \quad (9b)$$

with $h_i(t)=1$, if $\sigma(t)=i$ or $h_i(t)=0$, otherwise.

Substituting (9) into the cost (8), we can represent (6b) as

$$\sum_{t=1}^{k-T}\left\|y(t)-\Theta_{\sigma(t)}\begin{bmatrix}1\\ x(t)\end{bmatrix}\right\|^2=\mathrm{tr}\left(\sum_{i=1}^{s}\Theta_iH_i(k-T)\Theta_i'\right) -2\mathrm{tr}\left(\sum_{i=1}^{s}\Theta_iF_i(k-T)\right)+\mathrm{tr}\left(\sum_{t=1}^{k-T}y(t)y(t)'\right). \quad (10)$$

Note that the matrices $H_i(k-T)$ and $F_i(k-T)$ can be recursively computed as

$$H_i(k-T)=H_i(k-T-1)+x(k-T)x(k-T)'h_i(k-T), \quad (11a)$$

$$F_i(k-T)=F_i(k-T-1)+x(k-T)y(k-T)'h_i(k-T). \quad (11b)$$

Thus, as the new observation $\{x(k),y(k)\}$ is available, it is just required to update the matrices $H_i(k-T-1)$ and $F_i(k-T-1)$ through (11), without storing the time history of observations $\{x(k),y(k)\}_{t=1}^{k-T-1}$.

MIQP formulation

To reformulate (6) as an MIQP problem, let us define the vector $z_i(k)\in\mathbb{R}^{n_y}$ with $\delta_i(k)\in\{0,1\}$ as

$$z_i(k)=\big(y(k)-\Theta_ix(k)\big)\,\delta_i(k). \quad (12a)$$

Let M and m be an arbitrary large (resp. small) upper (resp. lower) bound of the elements of the vector $y(k)-\Theta_ix(k)$,

$$m\le y(k)-\Theta_ix(k)\le M. \quad (12b)$$

Based on conditions (10) and (12), problem (6) can be equivalently written as the MIQP problem

$$\min_{\substack{\{\Theta_i\}_{i=1}^{s}\\ \{\delta_i(k-t)\}_{i=1,t=0}^{s,T-1}\\ \{z_i(k-t)\}_{i=1,t=0}^{s,T-1}}} \sum_{t=0}^{T-1}\sum_{i=1}^{s}z_i^2(k-t)+ \quad (13a)$$

$$\mathrm{tr}\left(\sum_{i=1}^{s}\Theta_iH_i(k-T)\Theta_i'\right)-2\mathrm{tr}\left(\sum_{i=1}^{s}\Theta_iF_i(k-T)\right) \quad (13b)$$

$$+\sum_{t=0}^{T-1}\sum_{i=1}^{s}\left\|\big(p(k-t)-c_i\big)\delta_i(k-t)\right\|^2, \quad (13c)$$

$$\text{s.t. } z_i(k-t)\le M\delta_i(k-t), \quad (13d)$$

$$z_i(k-t)\ge m\delta_i(k-t), \quad (13e)$$

$$z_i(k-t)\le y(k-t)-\Theta_ix(k-t)-m(1-\delta_i(k-t)), \quad (13f)$$

$$z_i(k-t)\ge y(k-t)-\Theta_ix(k-t)-M(1-\delta_i(k-t)), \quad (13g)$$

$$\sum_{i=1}^{s}\delta_i(k-t)=1,\ t=0,\dots,T-1, \quad (13h)$$

$$\delta_i(k-t)\in\{0,1\},\ i=1,\dots,s. \quad (13i)$$

Summary and iterative refinement

The steps described so far for active mode selection (namely, cluster assignment of $p(k)$) and for model parameters Θ_i estimation are summarized in Algorithm 1. At the beginning of Algorithm 1, a mini-batch identification problem is solved to estimate the sequence of active modes $\sigma(t)$ from time 1 up to time T and to assign the scheduling vectors $\{p(t)\}_{t=1}^{T}$ to the cluster $\{\mathcal{C}_{\sigma(t)}\}_{t=1}^{T}$ (steps 2-6). Then, the observations $\{x(k),y(k),p(k)\}$ are processed iteratively and the MIQP problem (13) is solved (step 7.2). Besides updating the model parameters Θ_i at each time k (stage 7.3), the active mode $\sigma(k)$ is estimated (stages 7.4-7.6), the scheduling vector $p(k)$ is consequently assigned to cluster $\mathcal{C}_{\sigma(k)}$ (stage 7.7) and the cluster's centroid $c_{\sigma(k)}$ is updated (step 7.8-7.9).

Note that, for the first few data samples (i.e., for $\bar{k}\ll N$), the initial estimates $\{\sigma(t)\}_{t=1}^{\bar{k}}$ of the sequence of active modes may be inaccurate, as the active modes and the model parameters Θ_i are estimated based on a very few observations. At the next time samples $k>\bar{k}$, the estimate of both the active modes and the model parameters Θ_i are affected by inaccurate estimate of the initial mode sequence $\{\sigma(t)\}_{t=1}^{\bar{k}}$. This is due to the fact that the regularization cost (6b) depends on the estimated sequence $\{\sigma(t)\}_{t=1}^{\bar{k}}$. The effect of the initial misclassification may be reduced by running Algorithm 1 multiple times, including the sequences of active modes estimated at the previous runs, in the regularization term (6b). More specifically, at the n_q-th run of Algorithm 1, the following cost at each time k is considered instead of the objective function (6a)-(6c):

Algorithm 1 Recursive clustering and model parameters estimation

Input: Observations sequence $\{x(k), y(k), p(k)\}_{k=1}^{N}$; number of modes s; horizon T, initial clusters $\mathcal{C}_i$ and centroids c_i.

1. **let** $H_i(0) \leftarrow 0$, $F_i(0) \leftarrow 0$, $\mathcal{C}_i \leftarrow \emptyset$, $i = 1, \dots, s$;
2. **let** $k \leftarrow T$;
3. **solve** the MIQP problem (13);
4. **let** $\{\delta_i^*(t)\}_{i=1,t=1}^{s,T}$ be the optimal parameters minimizing (13);
5. **for** $t = 1, \dots, T$ **do**
   5.1. **let** $i^*(t)$ be the index such that $\delta_{i^*}^*(t) = 1$;
   5.2. **let** $\sigma(t) \leftarrow i^*(t)$;
   5.3. **let** $\mathcal{C}_{\sigma(t)} \leftarrow \mathcal{C}_{\sigma(t)} \cup \{p(t)\}$;
6. **end for**
7. **for** $k = T+1, \dots, N$ **do**
   7.1. **update** the matrices $H_i(k-T)$ and $F_i(k-T)$ through (11);
   7.2. **solve** the MIQP problem (13);
   7.3. **let** $\Theta_i^*(k)$ be the optimal parameters minimizing (13), $i = 1, \dots, s$;
   7.4. **let** $\{\delta_i^*(k-t)\}_{t=0}^{T-1}$ be the optimal parameters minimizing (13), $i = 1, \dots, s$;
   7.5. **let** i^* be the index such that $\delta_{i^*}^*(k) = 1$;
   7.6. **let** $\sigma(k) \leftarrow i^*$;
   7.7. **let** $\mathcal{C}_{\sigma(k)} \leftarrow \mathcal{C}_{\sigma(k)} \cup p(k)$;
   7.8. **let** $\delta c_{\sigma(k)} \leftarrow \frac{1}{|\mathcal{C}_{\sigma(k)}|}(p(k) - c_{\sigma(k)})$;
   7.9. **update** the centroid $c_{\sigma(k)}$ of cluster $\mathcal{C}_{\sigma(k)}$
   $$c_{\sigma(k)} \leftarrow c_{\sigma(k)} + \delta c_{\sigma(k)};$$
8. **end for**;

Output: Estimated parameters $\Theta_1^*(N), \dots, \Theta_s^*(N)$; clusters $\mathcal{C}_1, \dots, \mathcal{C}_s$; sequence of active modes $\{\sigma(k)\}_{k=1}^{N}$.

$$\sum_{i=1}^{s}\sum_{t=0}^{T-1} \left\|\left(y(k-t) - \Theta_i x(k-t)\right)\delta_i(k-t)\right\|^2 + \tag{14a}$$

$$\sum_{t=1}^{k-T} \left\|y(t) - \Theta_{\sigma(t,n_q)} x(t)\right\|^2 + \tag{14b}$$

$$\sum_{q=1}^{n_q-1} \lambda^{n_q-q-1} \sum_{t=1}^{N-T} \left\|y(t) - \Theta_{\sigma(t,q)} x(t)\right\|^2 + \tag{14c}$$

$$\sum_{t=0}^{T-1}\sum_{i=1}^{s} \left\|\left(p(k-t) - c_i\right)\delta_i(k-t)\right\|^2, \tag{14d}$$

with $\sigma(t,q)$ $(q = 1, \dots, n_q)$ being the estimate of the active mode at time t obtained at the q-th run of Algorithm 1. Note that (14c) is a regularization term based on the past runs of Algorithm 1, while (14b) plays the same role of (6b), as it regularizes the parameters Θ_i based on the estimate $\{\sigma(t)\}_{t=1}^{k-T}$ obtained at the current run of Algorithm 1. The influence of the past runs of Algorithm 1 is controlled by including a forgetting factor $\lambda \in \mathbb{R} : 0 < \lambda \leq 1$ in (14c), which exponentially down-weights the estimates $\{\sigma(t,q)\}_{t=1}^{N}$ obtained at the previous runs.

Similar to the original regularization term (6b), the cost (14b)-(14c) can be also recursively updated as a new sample $\{x(k), y(k)\}$ is processed, without the need to store the whole time history of estimates $\{\sigma(k,q)\}_{k=1,q=1}^{N,n_q-1}$ obtained at the previous runs of Algorithm 1. As a matter of fact, the terms (14b)-(14c) can be written as

$$\left\{\mathrm{tr}\left(\sum_{i=1}^{s}\Theta_i H_i(k-T,n_q)\Theta_i'\right) - \tag{15a}$$

$$2\mathrm{tr}\left(\sum_{i=1}^{s}\Theta_i F_i(k-T,n_q)\right) + \tag{15b}$$

$$\mathrm{tr}\left(\sum_{t=1}^{k-T} y(t)y(t)'\right)\right\} + \tag{15c}$$

$$\sum_{q=1}^{n_q-1}\mathrm{tr}\left(\sum_{i=1}^{s}\Theta_i\lambda^{n_q-q-1}H_i(N-T,q)\Theta_i'\right) - \tag{15d}$$

$$2\sum_{q=1}^{n_q-1}\mathrm{tr}\left(\sum_{i=1}^{s}\Theta_i\lambda^{n_q-q-1}F_i(N-T,q)\right) + \tag{15e}$$

$$\sum_{q=1}^{n_q-1}\mathrm{tr}\left(\lambda^{n_q-q-1}\sum_{t=1}^{N-T} y(t)y(t)'\right), \tag{15f}$$

where $H_i(N-T,q)$ and $F_i(N-T,q)$ $(q = 1, \dots, n_q)$ are defined similarly to (9) and computed based on the estimates $\sigma(t,q)$ given by the q-th run of Algorithm 1. Specifically:

$$H_i(N-T,q) = \sum_{t=1}^{N-T} x(t)x(t)'h_i(t,q),$$
$$F_i(N-T,q) = \sum_{t=1}^{N-T} x(t)y(t)'h_i(t,q),$$

with $h_i(t,q) = 0$, if $\sigma(t,q) = i$ or $h_i(t,q) = 1$, otherwise.

When the pair $\{x(k), y(k)\}$ is processed, the matrices $H_i(k-T,n_q)$ and $F_i(k-T,n_q)$ in (15a)-(15b) can be recursively updated through (11), while only the matrices $H_i(N-T,q)$ and $F_i(N-T,q)$ (with $q = 1, \dots, n_q - 1$) are needed to construct the terms (15d)-(15e).

3.2 Partition of the scheduling set

The partition $\{\mathcal{P}_i\}_{i=1}^{s}$ of the scheduling variable set $\mathcal{P}$ can be computed along with the estimation of the model parameters $\{\Theta_i\}_{i=1}^{s}$ and the sequence of active modes $\{\sigma(k)\}_{k=1}^{N}$. This is done by separating the computed clusters $\{\mathcal{C}_i\}_{i=1}^{s}$ using linear multicategory discrimination. In the following paragraph, we briefly describe the algorithm proposed in (Breschi et al., 2016), which is suitable for both offline and online (i.e., recursive) computation of the scheduling variable partition.

Linear multicategory discrimination

According to the formulation introduced in (Bennett and Mangasarian, 1994), the linear multicategory discrimination problem is solved by searching for a convex piecewise affine separator function $\phi : \mathbb{R}^{n_p} \to \mathbb{R}$ discriminating between the clusters $\mathcal{C}_1, \dots, \mathcal{C}_s$. The separator function ϕ is defined as

$$\phi(p) = \max_{i=1,\dots,s}\left(\begin{bmatrix} p' & -1 \end{bmatrix}\begin{bmatrix}\omega^i \\ \gamma^i\end{bmatrix}\right), \tag{17}$$

where $\omega^i \in \mathbb{R}^{n_p}$ and $\gamma^i \in \mathbb{R}$ are the parameters to be computed. Let m_i denote the cardinality of the cluster $\mathcal{C}_i$ and let $M_i \in \mathbb{R}^{m_i \times n_p}$, for $i = 1, \dots, s$, which is obtained

by stacking the scheduling vectors $p(k)'$ belonging to $\mathcal{C}_i$ in its rows. If the clusters $\{\mathcal{C}_i\}_{i=1}^s$ are linearly separable, then the separator function ϕ satisfies the following conditions:

$$[M_i \ -\mathbf{1}_{m_i}] \begin{bmatrix} \omega^i \\ \gamma^i \end{bmatrix} \geq [M_i \ -\mathbf{1}_{m_i}] \begin{bmatrix} \omega^j \\ \gamma^j \end{bmatrix} + \mathbf{1}_{m_i}, \quad i,j = 1,\dots,s, \ i \neq j, \tag{18}$$

where $\mathbf{1}_{m_i}$ is an m_i-dimensional vector of ones. The piecewise-affine separator ϕ thus satisfies the conditions:

$$\begin{cases} \phi(p) = [p' \quad -1] \begin{bmatrix} \omega^i \\ \gamma^i \end{bmatrix}, \ \forall p \in \mathcal{C}_i, \quad i = 1,\dots,s \\ \phi(p) \geq [p' \quad -1] \begin{bmatrix} \omega^j \\ \gamma^j \end{bmatrix} + 1, \ \forall p \in \mathcal{C}_i, \ i \neq j \end{cases} \tag{19}$$

From (19), each polyhedron $\mathcal{P}_i$, $i = 1,\dots,s$ is defined as

$$\mathcal{P}_i = \left\{ p \in \mathbb{R}^{n_p} : [p' \quad -1] \begin{bmatrix} \omega^i - \omega^j \\ \gamma^i - \gamma^j \end{bmatrix} \geq 1, \ j = 1,\dots,s, \ j \neq i \right\}.$$

According to (Breschi et al., 2016), the parameters $\{\omega^i, \gamma^i\}_{i=1}^s$ are calculated by solving the convex optimization problem

$$\min_{\xi} \frac{\kappa}{2} \sum_{i=1}^{s} \left(\|\omega^i\|_2^2 + (\gamma^i)^2 \right) + \sum_{i=1}^{s} \sum_{\substack{j=1 \\ j \neq i}}^{s} \frac{1}{m_i} \left\| \left([\, M_i \ -\mathbf{1}_{m_i} \,] \begin{bmatrix} \omega^j - \omega^i \\ \gamma^j - \gamma^i \end{bmatrix} + \mathbf{1}_{m_i} \right)_+ \right\|_2^2, \tag{20}$$

with $\xi = \left[(\omega^1)' \ \dots \ (\omega^s)' \ \gamma^1 \ \dots \ \gamma^s \right]'$. Problem (20) minimizes the squared 2-norm of the violation of the inequalities (18). The regularization parameter $\kappa > 0$ makes the cost in (20) strongly convex. Problem (20) is solved through the *Regularized Piecewise-Smooth Newton* approach explained in (Breschi et al., 2016) and originally proposed in (Bemporad et al., 2015).

4. SIMULATION EXAMPLES

The performance of the proposed LPV identification approach is shown via two simulation examples. In both of the examples, the training outputs are corrupted by an additive zero-mean white noise e with Gaussian distribution. The effect of the noise e on the output signal is quantified through the Signal-to-Noise Ratio (SNR), defined as $\text{SNR} = 10 \log \frac{\sum_{k=1}^{N} (y(k) - e(k))^2}{\sum_{k=1}^{N} (e(k))^2}$. The model quality is assessed on a noise-free validation data set of length N_{val}, not used in the training phase. The performance is quantified in terms of *Best Fit Rate* (BFR) defined as: $\text{BFR} = \max \left\{ 1 - \sqrt{\frac{\sum_{k=1}^{N_{\text{val}}} (y(k) - \hat{y}(k))^2}{\sum_{k=1}^{N_{\text{val}}} (y(k) - \bar{y})^2}}, 0 \right\}$, with $\hat{y}(k)$ being the simulated model output and $\bar{y}$ being the sample mean of the output over the validation set.

4.1 Example 1

Consider the SISO-LPV ARX data generating system:

$$y(k) = a_1^{\text{o}}(p(k))y(k-1) + a_2^{\text{o}}(p(k))y(k-2) + b_1^{\text{o}}(p(k))u(k-1) + e(k).$$

The p-dependent coefficients $a_1^{\text{o}}(p(k))$, $a_2^{\text{o}}(p(k)$ and $b_1^{\text{o}}(p(k))$ are described by the nonlinear functions:

$$a_1^{\text{o}}(p(k)) = \begin{cases} -0.5, & \text{if } p(k) > 0.5 \\ -p(k), & \text{if } -0.5 \leq p(k) \leq 0.5 \\ 0.5, & \text{if } p(k) < -0.5 \end{cases}$$

$$a_2^{\text{o}}(p(k)) = p^3(k), \quad b_1^{\text{o}}(p(k)) = \sin(\pi p(k)).$$

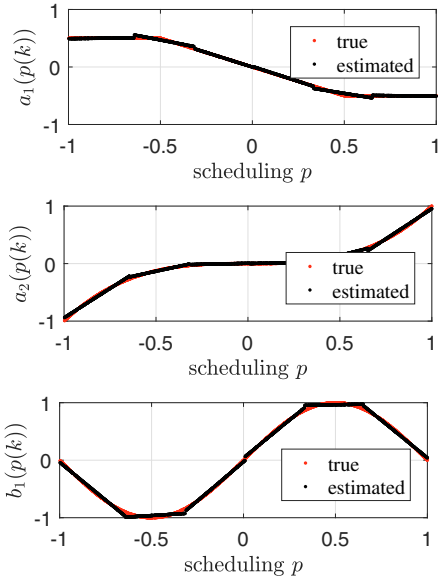

Fig. 1. Example 1. PWA estimates of the LPV coefficients A training data and a validation dataset of length $N = 6000$ and $N_{\text{val}} = 2000$, respectively, are generated. The training input u and the scheduling signal p are considered as independent white-noise processes with uniform distribution $\mathcal{U}(-1, 1)$. The standard deviation of the noise e is 0.05, which corresponds to SNR of 20 dB.

A PWA model with $s = 6$ modes is considered. Algorithm 1 is run with prediction horizon $T = 6$. Each MIQP sub-problem (13), solved with GPAD-B&B, contains 36 binary and 72 continuous variables, 144 inequality and 6 equality constraints. The average time to solve this problem is 0.13 sec for GPAD-B&B and 0.09 sec for GUROBI with default settings. In the second stage, off-line multicategory discrimination algorithm (Section 3.2) is executed for partitioning the scheduling space, and the estimated model parameters are refined based on the computed partition, using simple least-squares for each sub-model. The execution time to solve (20) is 5.6 sec. The BFR on the noise-free validation dataset is 0.95. The estimated LPV model coefficient functions are shown in Fig. 1.

4.2 Example2

Consider the MIMO LPV-ARX data generating system:

$$\begin{bmatrix} y_1(k) \\ y_2(k) \end{bmatrix} = \begin{bmatrix} \bar{a}_{1,1}(p(k)) & \bar{a}_{1,2}(p(k)) \\ \bar{a}_{2,1}(p(k)) & \bar{a}_{2,2}(p(k)) \end{bmatrix} \begin{bmatrix} y_1(k-1) \\ y_2(k-1) \end{bmatrix} + \begin{bmatrix} \bar{b}_{1,1}(p(k)) & \bar{b}_{1,2}(p(k)) \\ \bar{b}_{2,1}(p(k)) & \bar{b}_{2,2}(p(k)) \end{bmatrix} \begin{bmatrix} u_1(k-1) \\ u_2(k-1) \end{bmatrix} + e(k), \tag{21}$$

with

$$\bar{a}_{1,1}(p(k)) = \begin{cases} -0.3 & \text{if } \ 0.4\,(p_1(k) + p_2(k)) \leq -0.3, \\ 0.3 & \text{if } \ 0.4\,(p_1(k) + p_2(k)) \geq 0.3, \\ 0.4\,(p_1(k) + p_2(k)) & \text{otherwise}, \end{cases}$$

$$\bar{a}_{1,2}(p(k)) = 0.5\,(|p_1(k)| + |p_2(k)|)\,,$$

$$\bar{a}_{2,1}(p(k)) = p_1(k) - p_2(k),$$

$$\bar{a}_{2,2}(p(k)) = \begin{cases} 0.5 & \text{if } \ p_1(k) < 0, \\ 0 & \text{if } \ p_1(k) = 0, \\ -0.5 & \text{if } \ p_1(k) > 0, \end{cases}$$

$$\bar{b}_{1,1}(p(k)) = 3p_1(k) + p_2(k),$$

$$\bar{b}_{1,2}(p(k)) = \begin{cases} 0.5 & \text{if } \ 2\left(p_1^2(k) + p_2^2(k)\right) \geq 0.5, \\ 2\left(p_1^2(k) + p_2^2(k)\right) & \text{otherwise}, \end{cases}$$

$$\bar{b}_{2,1}(p(k)) = 2\sin\{p_1(k) - p_2(k)\}\,, \bar{b}_{2,2}(p(k)) = \ 0.$$

The training and the validation dataset consist of $N = 6000$ and $N_{\text{val}} = 2000$ samples, respectively. The in-

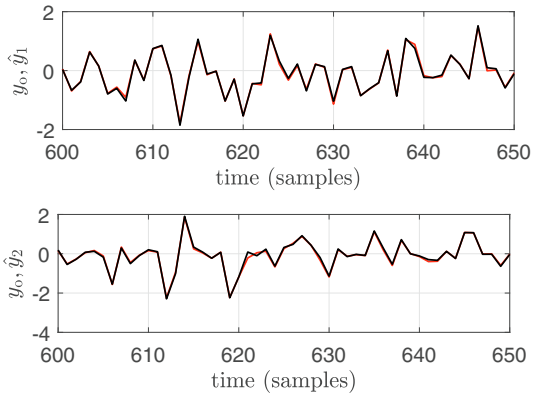

Fig. 2. Estimated (black) vs true (red) outputs

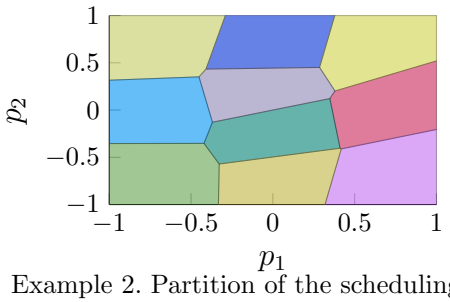

Fig. 3. Example 2. Partition of the scheduling space

Table 1. BFR on the validation data

runs(n_q)	BFR y_1	BFR y_2
1	0.8793	0.8108
2	0.8817	0.8146

put $u(k)$ and the scheduling vector $p(k)$ are white noise sequences generated drawn from a uniform distribution in the range $[-0.5 \quad 0.5] \times [-0.5 \quad 0.5]$ and $[-1 \quad 1] \times [-1 \quad 1]$, respectively. The covariance matrix of $e(k) \in \mathbb{R}^2$ is $\begin{bmatrix} 0.25e^{-2} & 0 \\ 0 & 0.25e^{-2} \end{bmatrix}$ which corresponds to signal-to-noise ratios on the first and on the second output channel equal to $\text{SNR}_1 = 23$ dB and $\text{SNR}_2 = 23.5$ dB, respectively. A PWA-LPV model with $s = 10$ modes is considered. Algorithm 1 is run for $n_\text{q} = 2$ iterations with the parameters: prediction horizon $T = 10$, forgetting factor $\lambda = 0.25$. In the second stage, problem (20) is solved to compute the partition of the scheduling space, with parameter $\kappa = 10^{-10}$ in (20). The obtained partition of the scheduling space is depicted in Fig. 3. The BFRs on the noise-free validation dataset are reported in Table 1 for both the output channels, and the output trajectories are shown in Fig. 2. For the sake of visualization, only part of the validation data are plotted. The obtained results show the capabilities of the estimated PWA-LPV model in reproducing the behaviour of the true LPV system.

5. CONCLUSIONS

The regularized moving-horizon piecewise-affine regression algorithm introduced in (Naik et al., 2017) has been properly adapted for the identification of *linear parameter-varying* (LPV) models, whose unknown coefficients depend in a *piecewise affine* (PWA) manner on the scheduling variable. Both the affine local functions and the partition of the scheduling-variable domain are directly estimated from data. This represents the main advantage of the approach with respect to parametric methods for LPV identification, which require to a priori chose basis functions describing the model coefficients. Current research activities are devoted to an automatic (namely, data-driven) selection of the LPV model structure, in terms of number of input lags, output lags and input delay.

ACKNOWLEDGEMENTS

The authors would like to thank Valentina Breschi for her help with the multicategory discrimination algorithm.

REFERENCES

Bamieh, B.A. and Giarré, L. (2002). Identification of linear parameter-varying models. *International Journal of Robust Nonlinear Control*, 12(9), 841–853.

Bemporad, A., Bernardini, D., and Patrinos, P. (2015). A convex feasibility approach to anytime model predictive control. Technical report, IMT Institute for Advanced Studies, Lucca. `http://arxiv.org/abs/1502.07974`.

Bennett, K. and Mangasarian, O. (1994). Multicategory discrimination via linear programming. *Optimization Methods and Software*, 3, 27–39.

Breiman, L. (1993). Hinging hyperplanes for regression, classification, and function approximation. *IEEE Transactions on Information Theory*, 39(3), 999–1013.

Breschi, V., Piga, D., and Bemporad, A. (2016). Piecewise affine regression via recursive multiple least squares and multicategory discrimination. *Automatica*, 73, 155–162.

Hsu, K., Vincent, T.L., and Poolla, K. (2008). Nonparametric methods for the identification of linear parameter varying systems. In *Proc. of the Int. Symposium on Computer-Aided Control System Design*, 846–851. San Antonio, Texas, USA.

Mejari, M., Piga, D., and Bemporad, A. (2016). Regularized least square support vector machines for order and structure selection of LPV-ARX models. In *Proc. of the 15th European Control Conf.*, 1649–1654. Aalborg, Denmark.

Mejari, M., Piga, D., and Bemporad, A. (2018). A bias-correction method for closed-loop identification of Linear Parameter-Varying systems. *Automatica*, 87, 128–141.

Naik, V.V. and Bemporad, A. (2017). Embedded mixed-integer quadratic optimization using accelerated dual gradient projection. In *Proc. 20th IFAC World Congress*, 10723–10728. Toulouse, France.

Naik, V.V., Mejari, M., Piga, D., and Bemporad, A. (2017). Regularized moving-horizon piecewise affine regression using mixed-integer quadratic programming. In *Proc. 25th Mediterranean Conf. on Control and Automation*, 1349–1354. Valletta, Malta.

Patrinos, P. and Bemporad, A. (2014). An accelerated dual gradient-projection algorithm for embedded linear model predictive control. *IEEE Trans. Automatic Control*, 59(1), 18–33.

Piga, D., Cox, P., Tóth, R., and Laurain, V. (2015). LPV system identification under noise corrupted scheduling and output signal observations. *Automatica*, 53, 329–338.

Piga, D. and Tóth, R. (2013). LPV model order selection in an LS-SVM setting. In *Proc. 52nd IEEE Conf. on Decision and Control*, 4128–4133. Florence, Italy.

Tóth, R., Laurain, V., Zheng, W.X., and Poolla, K. (2011). Model structure learning: A support vector machine approach for LPV linear-regression models. In *Proc. of the 50th IEEE Conf. on Decision and Control*, 3192–3197. Orlando, Florida (USA).